Can stable and accurate neural networks always be computed?

Matthew Colbrook (Cambridge, m.colbrook@damtp.cam.ac.uk)

Joint work with: **Vegard Antun** (Oslo), **Anders Hansen** (Cambridge)

Based on: M. Colbrook, V. Antun, A. Hansen, "Can stable and accurate neural networks be computed? - On the barriers of deep learning and Smale's 18th problem"

Code: www.github.com/Comp-Foundations-and-Barriers-of-AI/firenet

Interest in deep learning unprecedented and exponentially growing

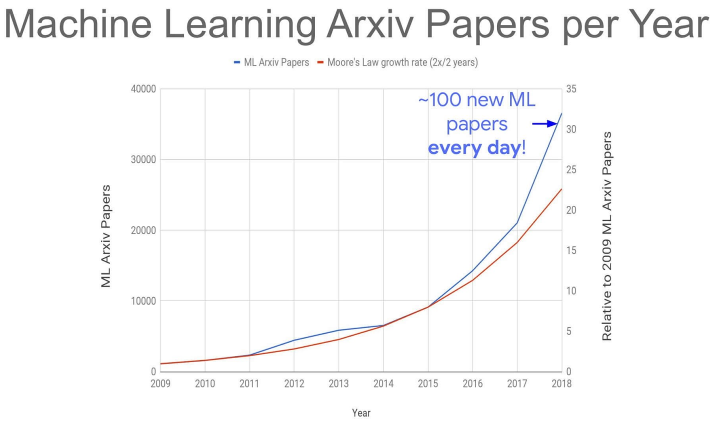


Figure: Source: ‘Deep Learning to Solve Challenging Problems’ (Google AI)

To keep up last year, you would need to continually read a paper every < 5 mins!

Will AI replace standard algorithms in medical imaging?

"superior immunity to noise and a reduction in reconstruction artefacts compared with conventional handcrafted reconstruction methods"

Published: 22 March 2018

Image reconstruction by domain-transform manifold learning

Bo Zhu, Jeremiah Z. Liu, Stephen F. Cauley, Bruce R. Rosen & Matthew S. Rosen

Nature **555**, 487–492(2018) | Cite this article

17k Accesses | **235** Citations | **197** Altmetric | Metrics

Abstract

Image reconstruction is essential for imaging applications across the physical and life sciences, including optical and radar systems, magnetic resonance imaging, X-ray computed tomography, positron emission tomography, ultrasound imaging and radio astronomy[1,2,3]. During image

Editorial Summary

Machine learning improves image reconstruction

Reconstructing images from data, whether for medical or astronomical purposes, hinges on well-defined steps. The data sensor encodes an intermediate representation of the observed

show all

DL is unstable in inverse problems!

PHYSICAL SCIENCES

On instabilities of deep learning in image reconstruction and the potential costs of AI

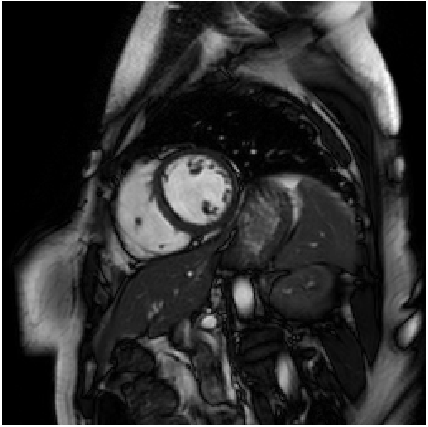
Vegard Antun, Francesco Renna, Clarice Poon, Ben Adcock, and Anders C. Hansen

PNAS first published May 11, 2020 https://doi.org/10.1073/pnas.1907377117

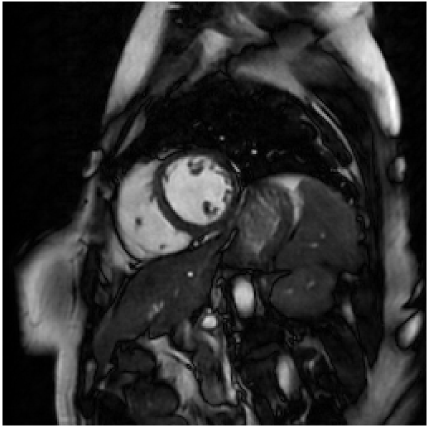
Edited by David L. Donoho, Stanford University, Stanford, CA, and approved March 12, 2020 (received for review June 4, 2019)

Example

$|x|$



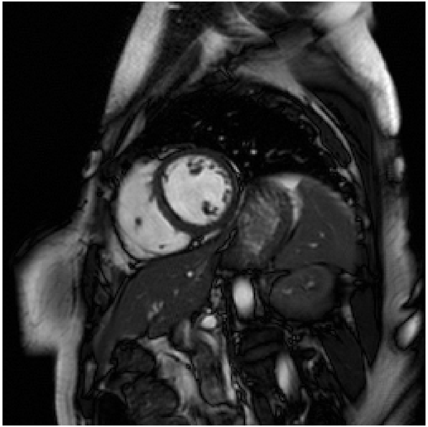
$|\Psi(Ax)|$



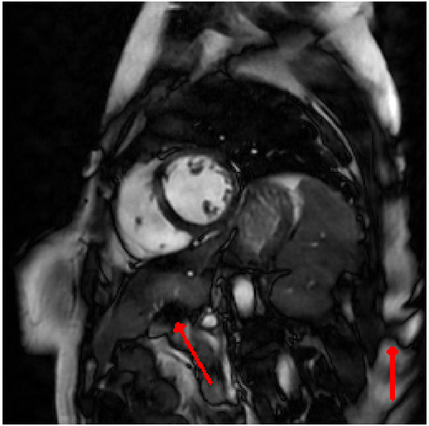
Network (33% subsampling) from: J. Schlemper, J. Caballero, J. V. Hajnal, A. Price and D. Rueckert, '*A deep cascade of convolutional neural networks for MR image reconstruction*', in International conference on information processing in medical imaging, Springer, 2017, pp. 647–658.
Figures from: Antun, V., Renna, F., Poon, C., Adcock, B., & Hansen, A. C., '*On instabilities of deep learning in image reconstruction and the potential costs of AI*'. Proc. Natl. Acad. Sci. USA, 2020..

Example

$|x + r_1|$



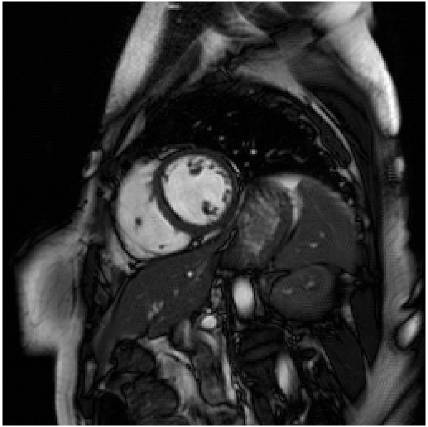
$|\Psi(A(x + r_1))|$



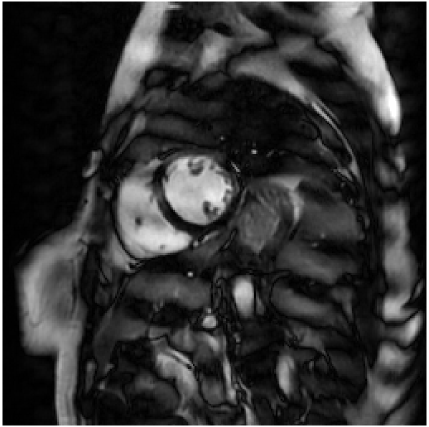
Network (33% subsampling) from: J. Schlemper, J. Caballero, J. V. Hajnal, A. Price and D. Rueckert, '*A deep cascade of convolutional neural networks for MR image reconstruction*', in International conference on information processing in medical imaging, Springer, 2017, pp. 647–658.
Figures from: Antun, V., Renna, F., Poon, C., Adcock, B., & Hansen, A. C., '*On instabilities of deep learning in image reconstruction and the potential costs of AI*'. Proc. Natl. Acad. Sci. USA, 2020..

Example

$|x + r_2|$



$|\Psi(A(x + r_2))|$

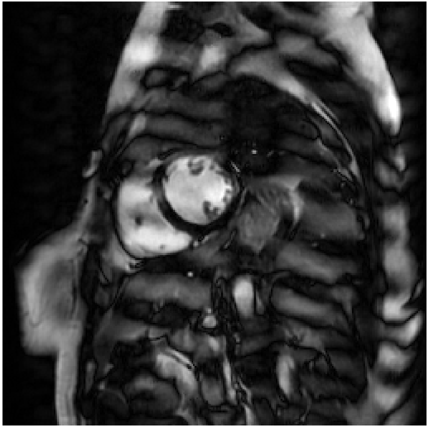


Network (33% subsampling) from: J. Schlemper, J. Caballero, J. V. Hajnal, A. Price and D. Rueckert, '*A deep cascade of convolutional neural networks for MR image reconstruction*', in International conference on information processing in medical imaging, Springer, 2017, pp. 647–658.
Figures from: Antun, V., Renna, F., Poon, C., Adcock, B., & Hansen, A. C., '*On instabilities of deep learning in image reconstruction and the potential costs of AI*'. Proc. Natl. Acad. Sci. USA, 2020..

Example

$\left|x + r_3\right|$

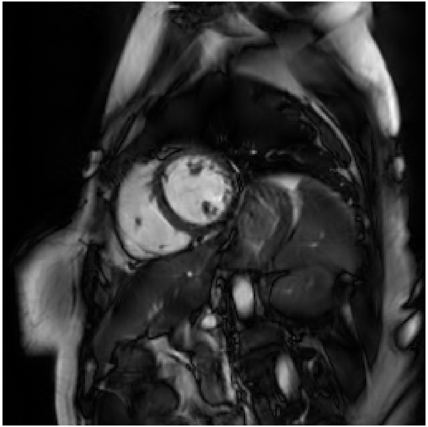
$\left|\Psi(A(x + r_3))\right|$



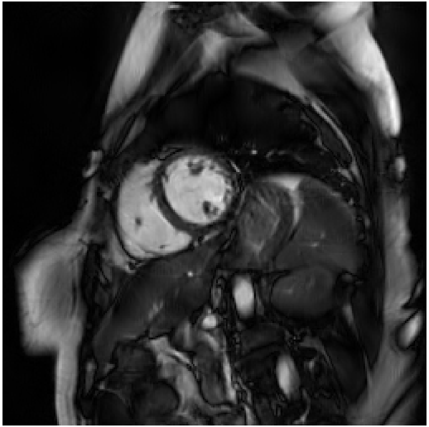
Network (33% subsampling) from: J. Schlemper, J. Caballero, J. V. Hajnal, A. Price and D. Rueckert, '*A deep cascade of convolutional neural networks for MR image reconstruction*', in International conference on information processing in medical imaging, Springer, 2017, pp. 647–658.
Figures from: Antun, V., Renna, F., Poon, C., Adcock, B., & Hansen, A. C., '*On instabilities of deep learning in image reconstruction and the potential costs of AI*'. Proc. Natl. Acad. Sci. USA, 2020..

Reconstruction using state-of-the-art standard methods

SoA from Ax



SoA from $A(x + r_3)$



Facebook and NYU's 2020 FastMRI challenge

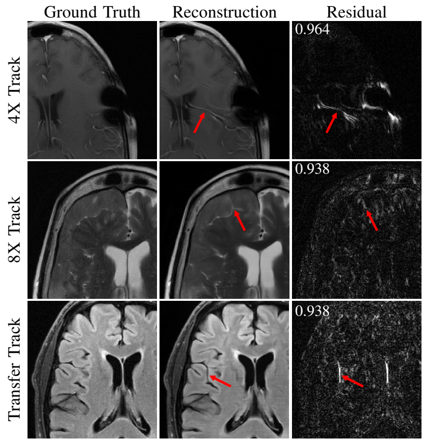


Fig. 6. Examples of reconstruction hallucinations among challenge submissions with SSIM scores over residual plots (residuals magnified by 5). (*top*) A 4X submission from Neurospin generated a false vessel, possibly related to susceptibilities introduced by surgical staples. (*middle*) An 8X submission from ATB introduced a linear bright signal mimicking a cleft of cerebrospinal fluid, as well as blurring of the boundaries of the extra-axial mass. (*bottom*) A submission from ResoNNance introduced a false sulcus or prominent vessel.

A program for the foundations of DL and AI

Smale's 18th problem*: *What are the limits of artificial intelligence?*

A program determining the foundations/limitations of deep learning and AI is needed:

- Boundaries of methodologies.
- Universal/intrinsic boundaries (e.g. no algorithm can do it).

There is a key difference between existence and construction here.

Need to also incorporate two pillars of numerical analysis:

- Stability
- Accuracy

GOAL for rest of talk: Develop some results in this direction for inverse problems.

*Steve Smale composed a list of problems for the 21st century in reply to a request of Vladimir Arnold inspired by Hilbert's list.

Mathematical setup

Given measurements $y = Ax + e$ recover $x \in \mathbb{C}^N$.

- $x \in \mathbb{C}^N$ be an unknown vector,
- $A \in \mathbb{C}^{m \times N}$ be a matrix $(m < N)$ describing modality (e.g. MRI), and
- $y = Ax + e$ the noisy measurements of x.

Outline:

- Fundamental barriers
- Sufficient conditions and Fast Iterative REstarted NETworks (FIRENETs)
- Balancing stability and accuracy

Can we compute neural networks that solve (P_j)?

Sparse regularisation (benchmark method):

$$\min_{x \in \mathbb{C}^N} \|x\|_{l^1} \quad \text{subject to} \quad \|Ax - y\|_{l^2} \leq \eta \tag{P_1}$$

$$\min_{x \in \mathbb{C}^N} \lambda \|x\|_{l^1} + \|Ax - y\|_{l^2}^2 \tag{P_2}$$

$$\min_{x \in \mathbb{C}^N} \lambda \|x\|_{l^1} + \|Ax - y\|_{l^2} \tag{P_3}$$

Denote the **minimising** vectors by Ξ.

- Avoid bizarre, unnatural & pathological mappings: (P_j) well-understood & well-used!
- Simpler solution map than inverse problem $\Rightarrow$ stronger impossibility results.
- DL has also been used to speed up sparse regularisation and tackle (P_j).

The set-up

$$A \in \mathbb{C}^{m \times N} \text{ (modality)}, \quad \mathcal{S} = \{y_k\}_{k=1}^{R} \subset \mathbb{C}^m \text{ (samples)}, \quad R < \infty$$

Question: Given a collection Ω of $(A, \mathcal{S})$, does there exist a neural network approximating Ξ (solution map of (P_j)), and can it be trained by an algorithm?

In practice, the matrix A is not known exactly or cannot be stored to infinite precision.

Assume access to: $\{y_{k,n}\}_{k=1}^{R}$ and A_n (rational approximations, e.g. floats) such that

$$\|y_{k,n} - y_k\| \le 2^{-n}, \quad \|A_n - A\| \le 2^{-n}, \quad \forall n \in \mathbb{N}.$$

And $\{x_{k,n}\}_{k=1}^{R}$ such that $\inf_{x^* \in \Xi(A_n, y_{k,n})} \|x_{k,n} - x^*\| \le 2^{-n}, \quad \forall n \in \mathbb{N}.$

Training set associated with $(A, \mathcal{S}) \in \Omega$ is

$$\iota_{A,\mathcal{S}} := \{(y_{k,n}, A_n, x_{k,n}) \mid k = 1, \dots, R, \text{ and } n \in \mathbb{N}\}.$$

What could go wrong?

$$\min_{x \in \mathbb{C}^N} \|x\|_{l^1} \quad \text{subject to} \quad \|Ax - y\|_{l^2} \leq \eta \tag{P_1}$$

$$\min_{x \in \mathbb{C}^N} \lambda \|x\|_{l^1} + \|Ax - y\|_{l^2}^2 \tag{P_2}$$

$$\min_{x \in \mathbb{C}^N} \lambda \|x\|_{l^1} + \|Ax - y\|_{l^2} \tag{P_3}$$

(i) There does not exist a neural network that approximates the function we are interested in.

(ii)

(iii)

What could go wrong?

$$\min_{x \in \mathbb{C}^N} \|x\|_{l^1} \quad \text{subject to} \quad \|Ax - y\|_{l^2} \leq \eta \tag{P_1}$$

$$\min_{x \in \mathbb{C}^N} \lambda \|x\|_{l^1} + \|Ax - y\|_{l^2}^2 \tag{P_2}$$

$$\min_{x \in \mathbb{C}^N} \lambda \|x\|_{l^1} + \|Ax - y\|_{l^2} \tag{P_3}$$

(i) ~~There does not exist a neural network that approximates the function we are interested in.~~

(ii)

(iii)

What could go wrong?

$$\min_{x \in \mathbb{C}^N} \|x\|_{l^1} \quad \text{subject to} \quad \|Ax - y\|_{l^2} \leq \eta \tag{P_1}$$

$$\min_{x \in \mathbb{C}^N} \lambda\|x\|_{l^1} + \|Ax - y\|_{l^2}^2 \tag{P_2}$$

$$\min_{x \in \mathbb{C}^N} \lambda\|x\|_{l^1} + \|Ax - y\|_{l^2} \tag{P_3}$$

(i) ~~There does not exist a neural network that approximates the function we are interested in.~~

(ii) There does exist a neural network that approximates the function, however, there does not exist an algorithm that can construct the neural network.

(iii)

What could go wrong?

$$\min_{x \in \mathbb{C}^N} \|x\|_{l^1} \quad \text{subject to} \quad \|Ax - y\|_{l^2} \leq \eta \tag{P_1}$$

$$\min_{x \in \mathbb{C}^N} \lambda \|x\|_{l^1} + \|Ax - y\|_{l^2}^2 \tag{P_2}$$

$$\min_{x \in \mathbb{C}^N} \lambda \|x\|_{l^1} + \|Ax - y\|_{l^2} \tag{P_3}$$

(i) ~~There does not exist a neural network that approximates the function we are interested in.~~

(ii) There does exist a neural network that approximates the function, however, there does not exist an algorithm that can construct the neural network.

(iii) There does exist a neural network that approximates the function, and an algorithm to construct it. However, the algorithm will need prohibitively many samples.

Bad news - can't necessarily approximate such a neural network

Theorem

For (P_j), $N \geq 2$ *and* $m < N$. *Let* $K > 2$ *be a positive integer,* $L \in \mathbb{N}$. *Then there exists a* **well-conditioned** *class (condition numbers* ≤ 1*)* Ω *of elements* $(A, \mathcal{S})$ *s.t. (*Ω **fixed** *in what follows):*

(i) *There* **does not exist any algorithm** *that, given a training set* $\iota_{A,\mathcal{S}}$, *produces a neural network* $\phi_{A,\mathcal{S}}$ *with*

$$\min_{y \in \mathcal{S}} \inf_{x^* \in \Xi(A,y)} \|\phi_{A,\mathcal{S}}(y) - x^*\|_{l^2} \leq 10^{-K}, \quad \forall (A, \mathcal{S}) \in \Omega. \tag{1}$$

Furthermore, for any $p > 1/2$, **no probabilistic algorithm** *can produce a neural network* $\phi_{A,\mathcal{S}}$ *such that (1) holds with probability at least* p.

(ii) *There* **exists an algorithm** *that produces a neural network* $\phi_{A,\mathcal{S}}$ *such that*

$$\max_{y \in \mathcal{S}} \inf_{x^* \in \Xi(A,y)} \|\phi_{A,\mathcal{S}}(y) - x^*\|_{l^2} \leq 10^{-(K-1)}, \quad \forall (A, \mathcal{S}) \in \Omega.$$

However, for any such algorithm (even probabilistic), $M \in \mathbb{N}$ *and* $p \in \left[0, \frac{N-m}{N+1-m}\right)$, *there exists a training set* $\iota_{A,\mathcal{S}}$ *such that for all* $y \in \mathcal{S}$,

$$\mathbb{P}\left(\inf_{x^* \in \Xi(A,y)} \|\phi_{A,\mathcal{S}}(y) - x^*\|_{l^2} > 10^{1-K} \text{ or size of training data needed} > M \right) > p.$$

(iii) *There* **exists an algorithm** *using only* L *training data from each* $\iota_{A,\mathcal{S}}$ *that produces a neural network* $\phi_{A,\mathcal{S}}(y)$ *such that*

$$\max_{y \in \mathcal{S}} \inf_{x^* \in \Xi(A,y)} \|\phi_{A,\mathcal{S}}(y) - x^*\|_{l^2} \leq 10^{-(K-2)}, \quad \forall (A, \mathcal{S}) \in \Omega.$$

In words...

Nice classes Ω where one can prove NNs with great approximation qualities exist. But:

- No algorithm, even randomised can train (or compute) such a NN accurate to K digits with probability greater than $1/2$.
- There exists a deterministic algorithm that computes a NN with $K-1$ correct digits, but any such (even randomised) algorithm needs arbitrarily many training data.
- There exists a deterministic algorithm that computes a NN with $K-2$ correct digits using no more than L training samples.

Result **independent of neural network architecture** - a universal barrier.

Existence vs computation (universal approximation/interpolation theorems **not** enough).

Conclusion: Theorems on existence of neural networks may have little to do with the neural networks produced in practice.

Numerical example: fails with training methods

$\mathrm{dist}(\Psi_{A_n}(y_n), \Xi_3(A,y))$	$\mathrm{dist}(\Phi_{A_n}(y_n), \Xi_3(A,y))$	$\|A_n - A\| \leq 2^{-n}$ $\|y_n - y\|_{l^2} \leq 2^{-n}$	10^{-K}	Ω_K
0.2999690	0.2597827	$n = 10$	10^{-1}	$K = 1$
0.3000000	0.2598050	$n = 20$	10^{-1}	$K = 1$
0.3000000	0.2598052	$n = 30$	10^{-1}	$K = 1$
0.0030000	0.0025980	$n = 10$	10^{-3}	$K = 3$
0.0030000	0.0025980	$n = 20$	10^{-3}	$K = 3$
0.0030000	0.0025980	$n = 30$	10^{-3}	$K = 3$
0.0000030	0.0000015	$n = 10$	10^{-6}	$K = 6$
0.0000030	0.0000015	$n = 20$	10^{-6}	$K = 6$
0.0000030	0.0000015	$n = 30$	10^{-6}	$K = 6$

Table: (**Impossibility of computing the existing neural network to arbitary accuracy**). A constructed from discrete cosine transform, $R = 8000$, $N = 20$, $m = 19$, solutions are 6-sparse. We demonstrate the impossibility statement (i) on FIRENETs Φ_{A_n}, and LISTA (learned iterative shrinkage thresholding algorithm) networks Ψ_{A_n}. The table shows the shortest l^2 distance between the output from the networks, and the true minimizer of the problem (P_3), with $w_l = 1$ and $\lambda = 1$, for different values of n and K.

Can we avoid this?

$$\hat{x} = \operatorname{argmin} f(x), \quad f^* = \min f(x)$$

Question: Can we find 'good' input classes where

$$f(x) < f^* + \epsilon \implies \|x - \hat{x}\| \lesssim \epsilon$$

We shall see that the answer is yes!

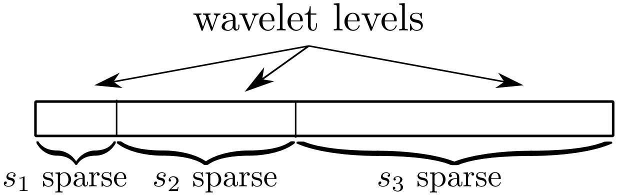
State-of-the-art model for sparse regularisation

Definition [Sparsity in levels]: Let $\mathbf{M} = (M_1, \ldots, M_r) \in \mathbb{N}^r$, where $1 \leq M_1 < \cdots < M_r = N$, and $\mathbf{s} = (s_1, \ldots, s_r) \in \mathbb{N}_0^r$, where $s_k \leq M_k - M_{k-1}$ for $k = 1, \ldots, r$ and $M_0 = 0$. A vector $x \in \mathbb{C}^N$ is $(\mathbf{s}, \mathbf{M})$-sparse in levels if

$$|\mathrm{supp}(x) \cap \{M_{k-1}+1, \ldots, M_k\}| \leq s_k, \quad k = 1, \ldots, r.$$

The total sparsity is $s = s_1 + \ldots + s_r$. We denote the set of $(\mathbf{s}, \mathbf{M})$-sparse vectors by $\Sigma_{\mathbf{s},\mathbf{M}}$. We also define the following measure of distance of a vector x to $\Sigma_{\mathbf{s},\mathbf{M}}$ by

$$\sigma_{\mathbf{s},\mathbf{M}}(x)_{l^1_w} = \inf\{\|x - z\|_{l^1_w} : z \in \Sigma_{\mathbf{s},\mathbf{M}}\}.$$



The robust nullspace property

Definition [weighted rNSP in levels]: Let $(\mathbf{s},\mathbf{M})$ be local sparsities and sparsity levels respectively. For weights $\{w_i\}_{i=1}^N$ ($w_i > 0$), we say that $A \in \mathbb{C}^{m\times N}$ satisfies the weighted robust null space property in levels (weighted rNSPL) of order $(\mathbf{s},\mathbf{M})$ with constants $0 < \rho < 1$ and $\gamma > 0$ if for any $(\mathbf{s},\mathbf{M})$ support set Δ,

$$\|x_\Delta\|_{l^2} \le \frac{\rho \|x_{\Delta^c}\|_{l^1_w}}{\sqrt{\xi}} + \gamma \|Ax\|_{l^2}, \qquad \text{for all } x \in \mathbb{C}^N.$$

$$\xi := \sum_{k=1}^{r} w_{(k)}^2 s_k, \quad \zeta := \min_{k=1,\dots,r} w_{(k)}^2 s_k, \quad \kappa := \frac{\xi}{\zeta}.$$

$$\text{rNSPL} \Rightarrow \|z_1 - z_2\|_{l^2} \lesssim \underbrace{\sigma_{\mathbf{s},\mathbf{M}}(z_2)_{l^1_w} + \|Az_2 - y\|_{l^2}}_{\text{“small”}}$$

$$+ \underbrace{\left(\lambda\|z_1\|_{l^1_w} + \|Az_1 - y\|_{l^2} - \lambda\|z_2\|_{l^1_w} - \|Az_2 - y\|_{l^2}\right)}_{F_3^A(z_1,y,\lambda) - F_3^A(z_2,y,\lambda)},$$

Main result

Simplified version of Theorem: *We provide an algorithm such that:*

Input: *Sparsity parameters* $(\mathbf{s}, \mathbf{M})$, *weights* $\{w_i\}_{i=1}^N$, $A \in \mathbb{C}^{m \times N}$ *(with the input* A *given by* $\{A_l\}$*) satisfying the rNSPL with constants* $0 < \rho < 1$ *and* $\gamma > 0$, $n \in \mathbb{N}$ *and positive* $\{\delta, b_1, b_2\}$.

Output: *A neural network* ϕ_n *with* $\mathcal{O}(n)$ *layers and the following property.*

For any $x \in \mathbb{C}^N$ *and* $y \in \mathbb{C}^m$ *with*

$$\underbrace{\sigma_{\mathbf{s},\mathbf{M}}(x)_{l^1_w}}_{\text{distance to sparse in levels vectors}} + \underbrace{\|Ax - y\|_{l^2}}_{\text{noise of measurements}} \lesssim \delta, \quad \|x\|_{l^2} \lesssim b_1, \quad \|y\|_{l^2} \lesssim b_2,$$

we have the following **stable** *and* **exponential convergence** *guarantee in* n

$$\|\phi_n(y) - x\|_{l^2} \lesssim \delta + e^{-n}.$$

Demonstration of convergence

Image | **Fourier Sampling** | **Walsh Sampling**

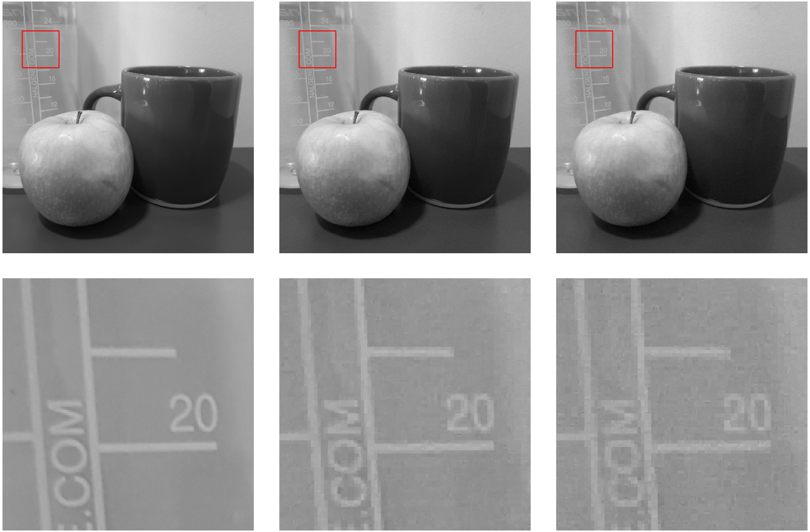
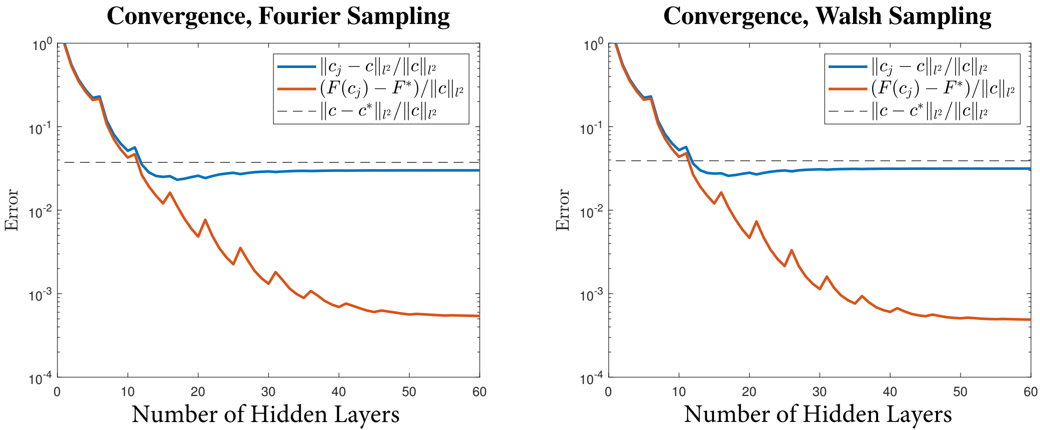
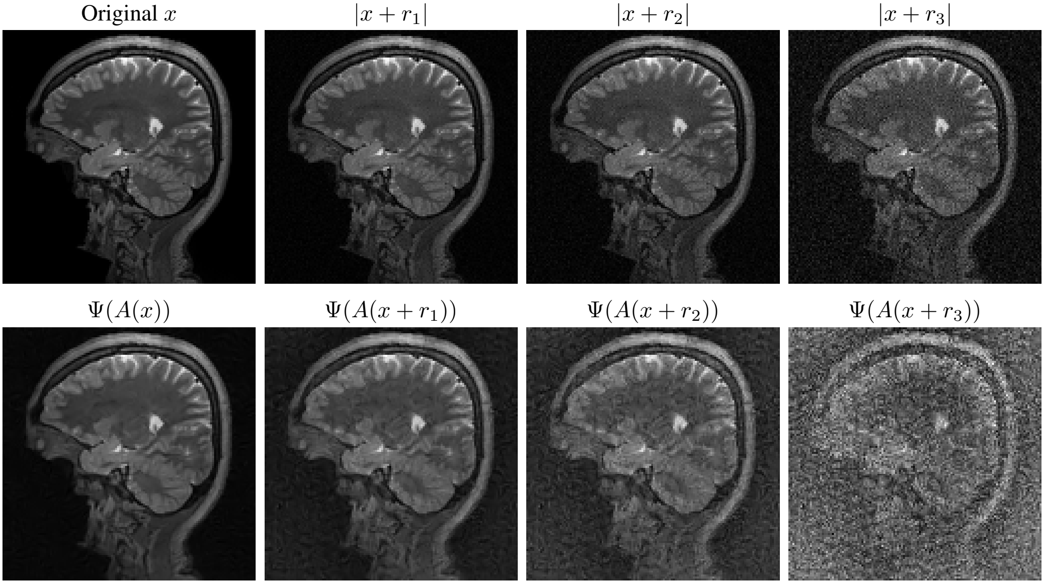


Figure: Images corrupted with 2% Gaussian noise and reconstructed using 15% sampling.

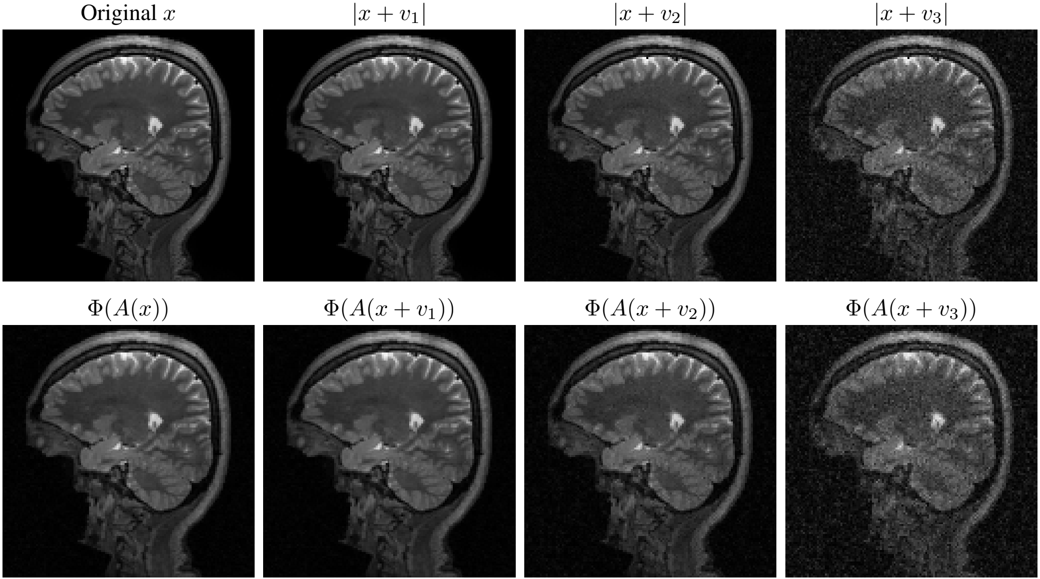
Demonstration of convergence



Stable? AUTOMAP ✗

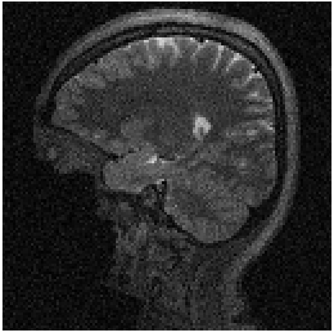


Stable? FIRENETs ✓



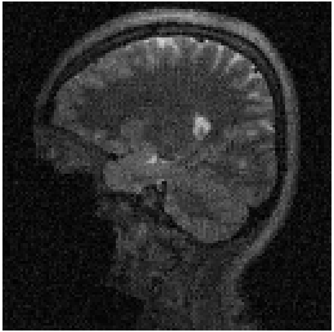
Adding FIRENET layers stabilises AUTOMAP

$|x + r_3|$



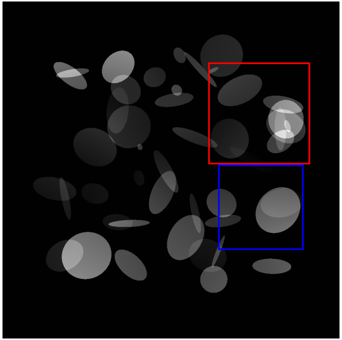
$\Psi(\tilde{y}),\ \tilde{y} = A(x + r_3)$

$\Phi\left(\tilde{y}, \Psi(\tilde{y})\right)$

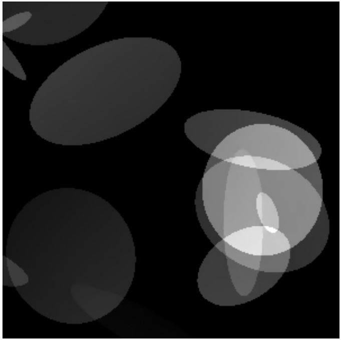


Stability and accuracy, and false negative

Original x
(full size)



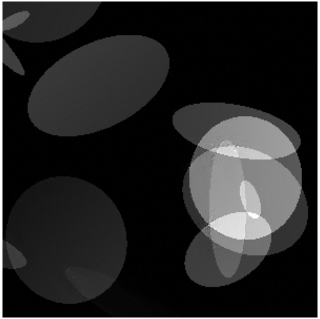
Original
(cropped, red frame)



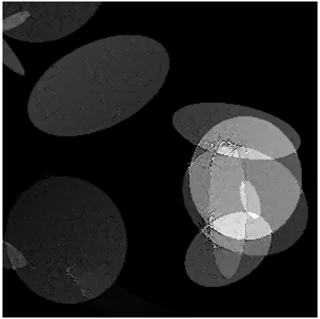
Original + detail ($x + h_1$)
(cropped, blue frame)

U-net trained without noise

Orig. + worst-case noise



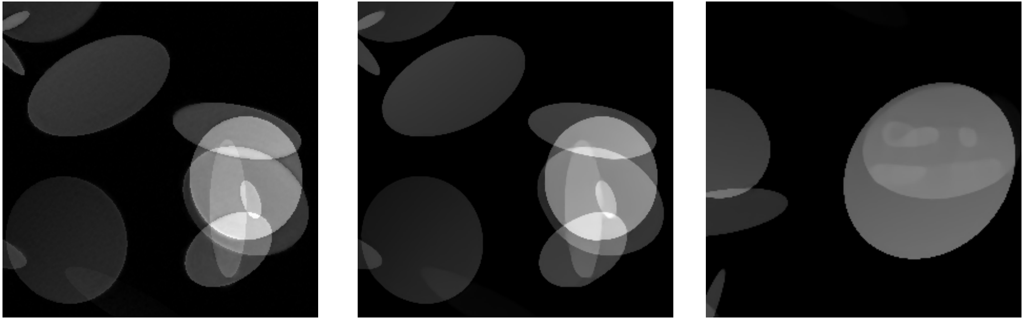
Rec. from worst-case noise



Rec. of detail

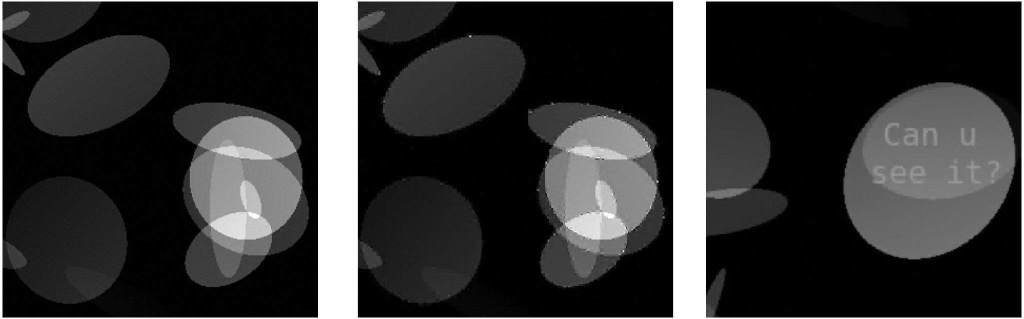
U-net trained with noise

Orig. + worst-case noise Rec. from worst-case noise Rec. of detail



FIRENET

Orig. + worst-case noise Rec. from worst-case noise Rec. of detail



Concluding remarks

There is a **need for foundations** in AI/deep learning. Our results:

- There are well-conditioned problems where mappings from training data to suitable NNs exist, but no training algorithm (even randomised) can approximate them.
- Existence of algorithms depends on desired accuracy. $\forall K \in \mathbb{Z}_{\geq 3}$, $\exists$ well-conditioned problems where simultaneously:
  (i) Algorithms may compute NNs to $K-1$ digits of accuracy, but not K.
  (ii) Achieving $K-1$ digits of accuracy requires arbitrarily many training data.
  (iii) Achieving $K-2$ correct digits requires only one training datum.
- Under specific conditions, there are algorithms that compute stable NNs. E.g., Fast Iterative REstarted NETworks (FIRENETs) converge exponentially in the number of hidden layers. We prove FIRENETs withstand adversarial attacks.
- There is a trade-off between stability and accuracy in deep learning.

Question: How do we optimally traverse the stability & accuracy trade-off? FIRENETs provide a balance but are likely not the end of the story.

Hopefully this talk has inspired you to build on these results and take up the challenge!